METHODOLOGICAL APPROACHES IN THE DEVELOPMENT OF PROGRAMS FOR GENERATING IMAGES

Bogdan Soban
Freelance generative artist, bogdan@soban-art.com, www.soban-art.com

Abstract
Methodological approaches in the development of programs for generating images

Creating images by the method of "generative art" is based on the fact that the authors themselves develop their software which provides an extraordinary diversity of the results. The approach to the development of programs can be highly mathematical, graphic design, or a combination of both. Authors have at disposal a number of known mathematical algorithms, or an immense field for innovative graphic manipulation, which enables them to achieve high level of diversity of the results. In the later period I have devoted a lot of effort just to this area and applied a number of methodological approaches which result in artistic features already hiding their computing origin. In this article I present five methodological approaches which often appear in my programs for generating images.

Key words: digital image, program, generate, algorithm, methodology, deformation, mutation, fractal, coloring palette

1. INTRODUCTION

The tendency of man to create a machine that would be similar to him is still remaining in the experiments. The information technology offers unprecedented possibilities for realization of this idea. Studies comparing the abilities of the computer and the human brain shows that tghe computer in some segments is far ahead: it is much faster in the implementation of mathematical and other operations, is infallible, it can permanently remembers the huge amount of data and it do not forget them, it is never tired, not emotionally burdened, independent of mood and generally not under the influence of others. Leveraging these characteristics is possible to simulate human creativity. We can speak about "Artificial Creativity" that gradually expands into the area of human creativity [1].

Within the field of artificial creation we find a creative method called "generative art" [2]. The most known definition of generative art was written by Philip Galanter, professor at New York University: “Generative art refers to any art practice where the artist creates a process, such as a set of natural language rules, a computer program, a machine, or other mechanism, which is the set to motion with some degree of autonomy contributing to or resulting in a complete work of art” [3]. The largest source of knowledge in this area is on the official website of the annual international conference organized by Professor Celestino Soddu from the Politecnico di Milano [4].

The basic characteristic of the generative approach is that the authors by the rule develop their own software which requires certain technical skills. In practice, such a person must be at first a computer programmer and then a painter. Developing own programs provides a diversity of styles of artworks. Most of the authors apply different iterative methods using known mathematical functions within their own software algorithms to create a recognizable image,

which can’t hide their computer origin [5].

In order to develop such software solutions whose results would be different from other artists I was looking for my own original approaches. The desire for diversity has led me to introduce my own methodological solutions in the programming. An interesting example of this kind was to use pre-generated images as color palette. Instead of pre-generated images is possible to use photos of an ambient to produce images that are switched on in the selected space. [6]

In terms of approach to the development of software for generating images they are known two basic principles: pragmatic and algorithmic approach. For the pragmatic method, which resulted in the abstract or figurative image type, is characterized that the author fairly accurate idea of the motive. To the program is left solely to decide on variations of the same theme. In algorithmic approach, however, the author is oriented much more into methodology than into final result of the process which is generally unpredictable. From the artistic point of view, those results are purely abstract and any association to objects from the real world is purely coincidental. Below I will describe my experience in the following methodological approaches: a pragmatic approach, multi-level algorithmic, diving into the picture and deformation approach.

2. MULTI-LEVEL ALGORITHMIC APPROACH

Z These kinds of programs are programs are characterized by multi-level algorithmic structure, which enhances significantly the number of types of images that the program can generate. A common weakness of programs to generate images is that the results are very similar or very low degree of the diversity of designs. Drawing capabilities of such a program are significantly better if generated motives are very different, and among them there are not many connections. It is the direction I follow when designing Creator, which is a typical representative of this multi-level algorithmic approach. The essential phases of the program are to define the genetic code of the future pictures, color algorithms (first level 240, the second level 20 color algorithms) and to draw a point on the screen in the selected color. The image emerges in a frontal mode (by columns and rows) or by randomly selected points. At the beginning of each program the genetic code is formed. It is composed of 60 variables which get random values. The second group of variables is generated from the actual process of drawing and the values of them currently change. These variables are produced by the calculation of distances and angles between current point and some other fixed poits on the screen. In subsequent solutions they were also included in the coordinates of points, which are controlled by multi-level circulation system. At the beginning of each cycle the path through the color algorithms is selected randomly. From the above, it is easy to calculate that there are 4,800 different channels and this means that the program knows the same number of different types of images. The values of the constants of the genetic code and the current values of the variables of the process enter into selected algorithm path. The result of the calculation are RGB components of the color of elaborated point. Example produced by this program is shown in Figure-01.

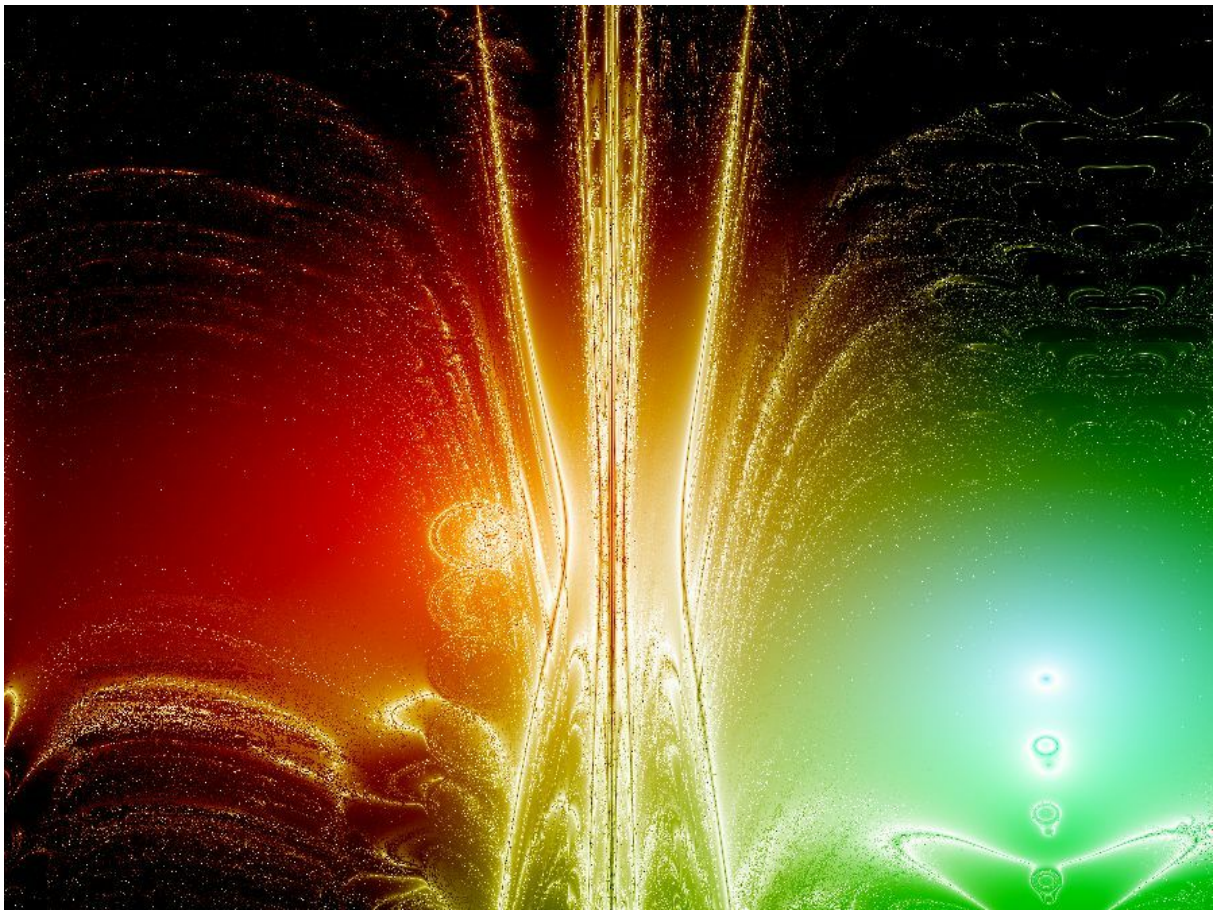

Figure-01

3. DIVING INTO PICTURE

Z In order to explore new approaches it has been developed a program DESIGN based on fractal calculus. Diving into the picture is a step in the direction to discover the opportunities offered by the integration of my own previous concepts with fractal technology. Fractal calculation works in a complex field and the iterations for each point of the complex plane verify its convergence or divergence. Based on the number of iterations the color of the point is determined. Study of fractal technology has opened up new possibilities, since it allows the zooming of the fractal images to nearly infinite dimensions. The process of zooming is manifested on the screen as a diving into the picture. Self-similarity of patterns at different depths as the basic characteristic of fractal didn't offer good motives. The problem was how to partially eliminate self-similarity. The solution is offered in the form of integration of my previous mathematical algorithms with fractal calculation. Practically using the variable depth in fractal calculus the distortion of the basic fractal logic has been done. The process wasn't any more a simple enlargements of selected portions of the image but it has been behaved as a traveling into third dimension of the image. The program is not any more fully automatic because it allows different selections from outside: color palette, image type and diving point. Example generated by this program is shown in Figure-02.

Figure-02

4. DEFORMATION METHOD

Deformation method is based on deformation of the previously created image, which typically consists of randomly selected geometric elements. The geometric elements are one, two and three-dimensional structures, painted on the basis of randomly selected color palette and nuanced. For the purposes of the study of this method it has been developed the program TEXTIMAGE. Geometric objects are of variable dimensions, which are randomly determined within selected areas. The size class area depends on the dimension of the image and it grows proportionally. At the very beginning of the cycle they are randomly selected types of objects, which will be performed in the current image. There is also selected the number of objects, position and the base color. Number of objects is determined to be approximately evenly fills the visual field. Special attention is paid to the color of objects. The method uses internal and/ or external color palette. [7]. Deformation of the image is carried out in two stages. In the first stage, the image is read by columns and rows, and on each point is performed deformation algorithm. In the second phase, the program reads the already partially deformed image by rows and columns and the further deform is done. Deformation algorithm is an essential part of this process. It can be described as follows: point currently processed has to be colored with the color of one of the points from the surrounding area. Which point from the surrounding area is selected it depends on a relatively simple calculation. The calculation is based on real-time generated variables inside the process. A detailed description of the deformation method was presented at the DSI in 2014 [8]. Example produced by this program

is shown in Figure-03.

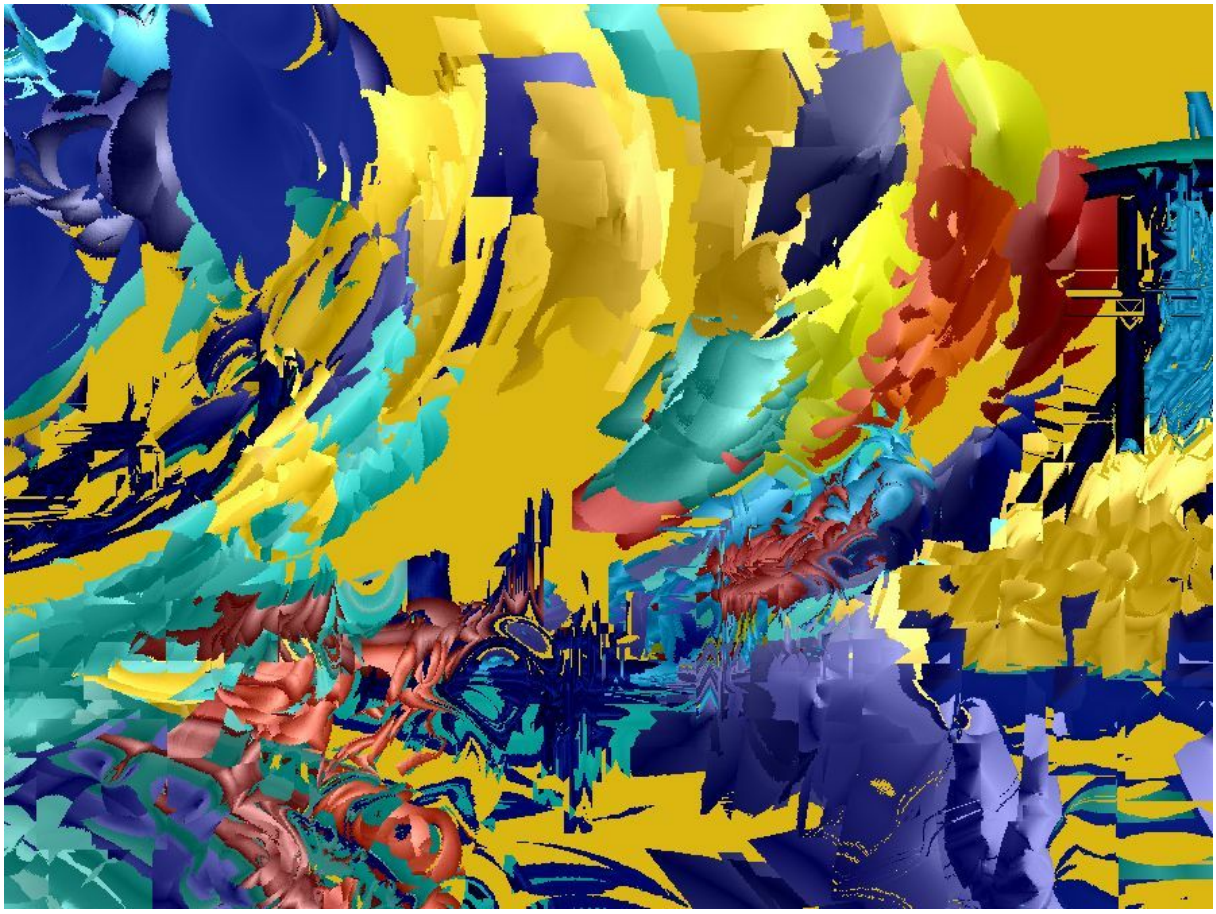

Figure-03

5. PRAGMATIC APPROACH

Pragmatic programming approach is described in the case of a program that generates autumn view of the Karst landscape, which is mainly due to the sumac beautiful colors. Such a process requires a detailed description of the Karst landscape elements, their possible deployment in the space and the appropriate color palette. For this purpose I developed the program KRAS. The program recognizes the following elements of Karst: tree, shrub, juniper, stone walls, rocks, grass, clouds and sky. Ale this elements are described as a kind of geometric shapes. All dimensions are variables that receive value only at the start of the programming cycle. The range of values for all dimension are consistent with the proportions of real Karst landscape. When the program starts, the type, number and location of objects in the image are defined. It is indicated the minimal number of objects because it is not necessary that all types are represented in one image. The arrangement of objects in space must respect the real configuration in the nature: bushes, grass and rocks below, the clouds and the sky above. Possible color shades for different types of objects are stored within the program in the color palette in the form of a matrix. The same object can take different colors, but inside its own color schemes. Partial effect of three-dimensionality is achieved by shading of individual objects. Drawing a picture takes place in three stages.

In the first stage a random selection of all the necessary parameters for a structure of the image is done. This in other words means that all variables that are located in the description of the objects obtained concrete value for the current programming cycle. In the second stage it draws a complete picture using geometric shapes filled with closed circles in the appropriate

Figure-04

colors. The picture looks as stylized geometric image of Karst. In the third stage it is used the algorithm for image processing which covers entire image with tiny irregular shapes which give the effect of brush. Example produced by this program is shown in Figure-04.

6 FREE DEMO SOFTWARE

All my programs are written in the programming language Visual Basic (VB6), which has high potential graphical programming. For those who want to try how this software works I developed demo versions of my original programs. Demo version of SW keeps all creative properties of original program, only the control possibilities from outside are limited. Using my demo SW is possible to create interesting images and to save them on computer as BMP files. Default folder to save images is C:\wimgx. For additional elaboration of generated images (resize and print) I recommend the use of any kind of graphics SW (IrfanView, Photoshop and similar). Demo software is possible to download from my web page having in consideration that is in the form of exe files which causes some problems with browser. Here

is the address of download page: www.soban-art.com/download.asp

The SW described in this paper the reader can find under following names:
CREATOR – for multi-level algorithmic approach
EXPLORING – for diving into picture
DEFORMATION – for deformation method
KRAS – for pragmatic approach
In the same page (www.soban-art.com/download.asp) there are other very simple generative programs as demo versions to make possible to understand how generative art works. Here is the complete list of SW programs with brief description:

DOTS – images composed by colored dots
LINES – images composed by colored lines
LINES-A – images composed by colored lines
RECTANGLES – images composed by colored rectangles
PATERNS – images composed by strong deformed circles
SYMMETRY – symmetric shapes
SYMMETRY-A – symmetric shapes
CURVES – typical algorithmic designed generative program
ROTATION – shapes composed by dots and lines
ROTATION-A – travelling line changing its color
CREATOR – real time image generator based on mathematical algorithms
MANDELBROT – explore the Mandelbrot fractal
EXPLORING – diving into third dimension of image
SYMMATRY-B – symmetric shapes (simple mandalas)
DEFORMATION – multilevel deformation approachB
SHAPELESS – color harmonized shapeless forms
GREETINGS – personalized greetings card based on your name
KRAS – Karst – typical Slovenian landscape, colorful in autumn
3D OBJECTS – semi-interactive pragmatic designed program (short code)
COLLAGE – program makes collage by clicking on the screen
FREEHAND – pure interactive designed program (freehand drawing)
SQUARES – typical pragmatic designed generative program
DISCOVERER – semi-interactive program discovering basic image depth
SPRAYBRUSH – image generator using spray technique
MANDALA – from order to chaos and back (symmetric image)
TEXTIMG – text based image generator
MUTATE – mutations of world famous artworks (needs internet connection)

7 CONCLUSION

Different authors use a large range of different approaches to develop generative programs. Authors who switched from classical painting to computer image generation is characterized a strong influence to the creative process and the results if it. Many of them therefore mainly use professional software and are less engage in their own development. For authors who

have switched from computer science to the field of art it is characterized that they build much more autonomous systems which could already meet the definition of "generative art". Here I have to mention Matjaz Hmeljak [9], Andrej Bauer [10] as two representatives of artificial art in my neighborhood.

The purpose of this paper is primarily to impress new authors for this kind of creative approach. Generative art becomes the world leads way in all areas of human creativity and a lot of new solutions realized in practice were obtained in this way. Of course, it is required knowledge of programming, but the method described above show that the development of such programs doesn't necessary requires high knowledge of mathematics. Usually it needs only some innovative ideas in the area of manipulation of pictures. As an interesting experiment in deepening this method I want to mention the thesis of David Rebernik from the Maribor Faculty of Education, Department of Fine Arts [11]. Maybe this contribution will challenge new authors to use a new creative approach which becomes creative paradigm of the future.

SOURCES AND LITERATURE

1. SAUNDERS, Rob: Artificial Creativity: a Syntetic approach to the Study of Creative Behavior, dostopno na internetu 10.6.2016, http://cs.gmu.edu/~jgero/publications/2001/SaundersGeroHI01.pdf
2. WIKIPEDIA: Generative art, dostopno na internetu 12.6.2016, , http://en.wikipedia.org/wiki/generative_art
3. GALANTER, Philip, What is Generative Art, dostopno na internetu 12.6.2016, http://www.philipgalanter.com/downloads/ga2003_paper.pdf
4. SODDU, Celestino: Generative Art International Conferences, dostopno na internetu 12.6.2016, www.generativeart.com
5. WATZ, Marius: Generator.x – Art from Code, Software and Generative Strategies in Art and Design, dostopno na internetu 12.6.2016 www.generatorx.no
6. SOBAN, Bogdan, Računalniško generiranje slik z uporabo predhodno ustvarjene slike kot barvne palete, Zbornik predavanj, Dnevi slovenske informatike DSI-2006
7. BARRALLO, Javier: Coloring Dynamical Systems, dostopno na internetu 15.6.2016, http://math.unipa.it/~grim/Jbarrallo.PDF
8. SOBAN, Bogdan, Generiranje slik z uporabo deformacijske metode, Zbornik predavanj, Dnevi slovenske informatike DSI-2014
9. HMELJAK, Matjaž: Matjaž Hmeljak – VISTRO 84 – 06, dostopno na internetu 15.6.2016, http://www.hmeljak.com
10. BAUER, Andrej: Random Art, dostopno na internetu 15.6.2016, http://www.random-art.org
11. REBERNIK, David, Sodobne umetniške prakse v delih Bogdana Sobana in lastnih delih, Diplomsko delo, Pedagoška fakulteta Maribor, 2011